Systematization and Clarification of Semantic Web Annotation Terminology

Nuno Bettencourt[1], Paulo Maio[1], András Pongó[2], Nuno Silva[1] and João Rocha[1]
Research Group on Knowledge Engineering and Decision Support
School of Engineering - Polytechnic Institute of Porto
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto - Portugal
[1]{nmgb,paom,nps,jsr}@isep.ipp.pt
[2]pa2@infolanc.hu

***Abstract*—This paper gives an overview of current semantic web annotation terminology and technology. For this, it focuses on the systematization of existing terms according to different contexts, projects and application areas. In order to present a clear definition of terms, scientific and commercial projects are related to the processes each one is expected to support in the semantic web annotation tasks.**

Keywords: Semantic Annotation, Ontology

I. Introduction

Ever since books exist, people felt the need to mark up what they read, whether to know were they stopped reading the last time or to give higher connotation to a single passage. There are several kinds of techniques to mark up documents, though highlighting, underlining, commenting, and writing between lines are probably the most used. Their application depends not only on the person doing annotation but also on the type of physical support (*e.g.* book, magazine, and newspaper), type of content (*e.g.* news, ads, romance, technical, raw data) and purpose of reading (*e.g.* work, entertainment) and purpose of annotation (*e.g.* learning, memorization, teaching, organization). For example, when someone is reading a review between two objects, some decide to underline disadvantages and highlight advantages while others decide to do exactly the opposite.

When web pages and electronic documents came along, some problems arose for people who were used to taking notes on books between lines, commenting/annotating on the margins or even using post-its. At that time people often decided to make hard-copies of the document and annotate using traditional methods. However, using hard-copies of an electronic document, the user looses the intrinsic advantage of this type of document, with respect to the continuous update of the document. In fact, once hard-copied, the hand-made annotations are no longer contextually valid in the electronic-based updated document.

Creating, maintaining and associating the annotations with the electronic versions of document are very important requirements. Some tools can help annotating e-documents (*e.g.* e-books, pdf documents) and web pages [13], easing the specification and maintenance of annotations along with the documents. Still, storage and sharing of any kind of document assumes such huge dimensions that new approaches are arising for many common-sense processes, exploiting new technologies and services. More recently, with the advent of the Semantic Web, machine processable data and semantic-based services are delivering, but also claiming, increasingly more semantically-rich data. Web services and agent-based systems are two of these paradigms. Annotations represented according to formal notations and conceptualization of knowledge domains (*e.g.* ontologies) are seen as one of the possibilities to relate semantics with online databases, services and online rendering web documents.

In the next section the annotation term is characterized and clarified into several variations. In particular, it focus on the distinction between annotation object and process, together with different perspectives of applications. The third section analyses multiple dimensions of the annotation object and process, relating difficulties with currently existing approaches. This section is the core of the paper both in extension and importance. The fourth section summarizes the content of previous sections, emphasising the problems and future research directions.

II. Variations On Annotation

Annotation refers to both the process and the object resulting from the process. The process and the object are so closely interrelated that in most cases it makes no sense to separate one definition from the other.

According to [6], the annotation object is the content represented in a formal language and attached to the document. This definition agrees with that of [12] when the authors state that "annotations are viewed as statements made [...] about a Web document". Thus, annotation is understood as an independent document, yet existing only in respect to the content of the document(s) it refers to.

Annotation is often related to the concept of meta-data and the processes of indexing and information retrieval (IR). Though meta-data is traditionally associated with the categorization and indexing of documents, it is no longer or never has been, fundamental for these tasks. In fact, indexing and IR engines exploit much more complex elements of the document, including the content of the document itself. Information extraction mechanisms (see III.L) and inference of inter-document relations based on the analysis of the user's browsing activities [2] are some of the currently more used approaches.

With the emergence of the semantic web and web services, semantically-rich descriptions of document content and services interfaces become a very important issue. Online rendered web documents based on hidden databases are now a very common way to provide web documents. In particular, e-commerce data sheets are commonly available, although they are generated mainly for human consumption. Automatic services however, require a complete different perspective of information in the web document, including the specification of data structure and semantics. When the annotation semantically enriches the content in a formal, machine readable way, it is referred to as Semantic Annotation. Semantic annotations together with ontologies are envisaged as being capable of providing these elements. This new perspective on semantic annotation is referred as "deep annotation".

As described, the concept of annotation significantly evolved during recent years, especially in the context of the web and the semantic web. Many technological solutions have been suggested to support the annotation process. As consequence, different approaches and perspectives of the annotation process and object have been used. The diversity of annotation process and object is sometimes responsible for the difficulty in understanding the various solutions, contributions and achievements.

III. Annotation Dimensions

This section contributes to the clarification of the knowledge domain of document annotation, by analysing and distinguishing the different dimensions of the annotation process and object. Through this section, several of these dimensions will be analysed and the most relevant work for each will be referenced. Though not specifically referring to the semantic web, the fact is that most of the analysis will focus primarily on the annotation on the web and semantic web.

A. Applications of Annotation

The application dimension is related to the (business) process in which the annotation object is used to accomplish the organization goals. While many different application scenarios exist, the most often recognized are:

- Document visualization, which refers to the exploitation of annotations by the user through document visualization tools. Underlining, cross-out, highlighting or inclusion of small signs (*e.g.* for editing purposes) are commonly exploited and useful in this context. Minimal or no automation is involved in this application scenario.

- Information indexing and retrieval concerns the use of annotations attached to the documents to improve both the indexing and retrieval capabilities of the service. Meta-data and ontology-based description of content is included in the annotations. This application scenario is mostly automated in both indexing and retrieval.

- Knowledge management is related to the capabilities of the organizations to acquire, store and exploit knowledge from its collaborators during business activities. Along with formal specification of annotations, representation of processes workflow associated with documents are often fundamental, which further includes the capability to relate many document and parts of documents. Automation of the process is required, but often considered difficult because of the insufficiency of either the acquisition and specification process, or the knowledge retrieval process.

- Business automation, namely by providing business competences through web services and agent-based systems. Annotations are seen as formal representation of business competences provided by the automation mechanism, which are published and advertised in the web through specialized services such as UDDI [19].

B. Annotation Types

The type of the annotation is concerned with the changes to the document content. The type is closely related to the application one does from it. The most relevant are those that:

- Mark up the document content, especially to visually stress out parts of the document;

- Relate documents or parts of document(s), in order to create (factual or mind-based) correlations between document(s) content;
- Semantically enrich the content of the document, such as formalizing its content for use by automatic processes;
- Complement the content of the document, such as corrections, update or addition of new content to the document.This purpose might be seen as a variation of previous two or even with document evolution and versioning.

The way an organization makes use of the different types of annotation is addressed in the application dimension. For example, a mark up annotation may be used by one organization for visualization purposes only, while another might use the same type to automatically evaluate the need for a document revision.

C. Annotation Models

Based on the type and application of the annotation and the domain of knowledge to represent, it is necessary to define the structure and semantics of the annotation (object), *i.e.* the model of the annotation. In the simplest cases, the model is very simple (*e.g.* one string, one URI), corresponding to "comment", "example", "explanation" or "see also" requirements. In more complex ones though, the annotation model should be specified in a way that it can be shared and understood among the information/knowledge community that use it. In some cases, machines are required to reason upon the model, requiring a higher level of formality and machine awareness.

In order to achieve formality and machine readability, ontologies are currently used widespreadly. Ontologies are much more than formal representation notations, since they describe domain (real-world) concepts in an unambiguous manner, providing the mechanisms to bind domain concepts to machine representable symbols (*e.g.* strings-based comments, relations between documents, images) [18].

D. Representation Language

The representation assumes a central role in the context of web and semantic web due to the need to store and share annotations across different systems. This is especially true when dealing with HTML documents, to which annotations are attached and simultaneously visualized by distinct browsing tools. Additionally, in deep annotation scenarios and business intelligence, annotations are usually consumed by automatic services, requiring *de facto* standard representation languages *e.g.* RDF[23].

Instead, though playing an important role in the interoperability process, it is not so relevant to use standard representation languages within closed or controlled environments and applications. This is the case of Adobe Acrobat and MS Word annotations. Besides, some of these proprietary tools often allow exporting annotations to other formats and representation languages, easing further interoperability.

E. Type of Document

Several types of documents are common on the web. HTML pages, PDF, DVI files, as well as images, movies and audio documents can be seen or downloaded from the internet and stored locally. Although all these types of objects are available for reading or viewing/listening on the web, web pages are the most common document type encountered, and are therefore the most common type supported by annotation tools.

Most annotation tools support only text-based documents [3, 4, 21, 9] but some support other types: static html documents [13], dynamic generated html documents from database [10], image annotation [11, 8], radio and tv news [17].

F. Rendering

Annotation rendering is the process whereby the annotations are embedded in the document to which they refer. Commonly, when the annotations are consumed by human-user, the visualization tool chosen is an important factor in the process. Instead, by the use of automatic processing mechanisms such as web services and agent-based systems, annotations are not presented to human users, but provided in a more machine-oriented notation (*e.g.* XML, RDF). Even though the description presented in the rest of this section focus on rendering HTML documents, two types of annotation rendering can be generalized for other types of documents and scenarios as well: client-side and server-side.

The client-side approach has been introduced by the Annotea project [12]. In general terms, there is a client application responsible for merging any annotation with a web page being browsed, showing the highlighted or underlined text as it is annotated. This kind of behaviour can be considered as a two step web page rendering approach. First, the original web page browsed by the user is provided by the server. Second, this web page is marked up on the client application (*e.g.* special web browser, or any browser plug-in) according to the annotations available or chosen by the final user.

In the server-side approach, web page annotations presented to the final user are embedded by the web server,

while rendering the web page. This type of rendering is normally proposed by proprietary integrated systems, which are responsible for storage (see III.H), query (see III.I) and embed the annotations onto the web document, such the client has no influence on the process. The annotation management process is provided by the server interface provide in the page itself. This approach is advantageous when client browser has no annotation competences.

Server annotation rendering approaches may have better rendering times, as the loaded annotations are local to the hosting server. This fact might not be totally true if for example, the web server uses other annotation servers. Client side approaches, on the other hand, also have some advantages over the server side approach, as it can have more available settings for managing and rendering pages. Especially, the client side approach only renders the web page with annotations if demanded by the user, avoiding new document content requests.

The annotations body can be embedded on the web page when rendered, or requested from the server as needed by the client browser. In the latter situation the web page is initially rendered with only the annotation meta-data necessary for its late retrieval from the server.

The presentation of the annotation in the visualization tool is very dependent on the tool and on the annotation model. Specifically, the anchor of the annotation is represented as a remark sign, that should be set according to the annotation type and the user preferences. The remark signs embedded in document provide the indication to the user about the existence of an annotation in the document. Additionally, because different signs are used, the type and model of the annotation are easily noticeable. Essentially, these remarks are used like the underline beneath hyperlinks, providing the evidence to the user that an image or text acts as a hyperlink to another web page.

Types or remarks only have meaning for interactive browsing and therefore are irrelevant to machines processing or consuming annotations automatically. Highlighting, underlining, balloons, geometric forms (*i.e.* circles, squares, rectangles) and other types of remarks can be used by the browser, defined by the user and assigned to an annotation to a web page. These symbols or ways of marking up documents should be customised by the user so that assignment and easier relations can be established between annotations and their display on the screen, enabling better page readability for the final user. The Annozilla [13] plugin for the Mozilla browser uses several, non customizable signs for several models of annotations.

G. Dynamicity of Annotation

Dynamicity is concerned with the nature of the annotation content. Two distinct kinds have been generalized:

- Static annotation is made of static content (*e.g.* text, URI, image). In this context it should be something that never changes or has few changes overtime. For example, if an annotation is made about someone, their birth date can be considered a static annotation as it will almost certainly never change. Annotations are traditionally static;

- Dynamic annotation, also called rule annotation, is obtained as a result of a query or filtering by an automatic process according to the content of the document and the data source of the annotation. Dynamic annotations do not exist by themselves but are achieved by conditions or rules (*e.g.* queries, filters).

Deep annotation is notably a case of dynamic annotation, in which the purpose of the annotation is to semantically enrich the document content. Typically, the document is created along with the semantic annotations from the database or knowledge base. When the query is performed, the results are used to render the document for human users, and to specify the content in a formal, semantically unambiguous way through semantic annotations.

H. Storage

Independently of other dimensions on the annotation process and object, annotation objects should be stored in order to be accessed in the future.

Annotations can be kept either on local or remote repositories. When annotations are stored in a local repository, *e.g.* local hard-drive, they can be called local annotations. When these annotations are saved on remote repositories like web servers, they are called remote annotations [12, 22]. Both local and remote annotations have advantages and disadvantages.

Local annotations are stored in a local repository, thus taking the advantage of easy security (see III.J) and easy specification of models, representation languages, types and applications (see above). Sharing however will be harder. On the contrary, remote annotations can be shared, published and read by any user that wishes to have access to them. Remote annotation systems are typically characterized by the use of standard representation languages and simple annotation models. This is due to the fact that it is difficult to agree about complex structures and their semantics (*e.g.* ontology models). This problem will be further addressed in III.I.

Questia [14] is an online library that makes use of remote annotations. These systems can offer very good features as search and filtering, since they have control over all the annotations made by its application. Nevertheless, these types of systems are normally proprietary and therefore it may become difficult to interact with other

annotation systems if they dont make use of standard annotation storage and specification. Another problem that is even more concerning is that one cannot rely on proprietary systems if they do not have any way of backing up annotations to the local computer. For example, if these kind of proprietary services cease to exist, one may loose all annotations made on some page which sometimes, can be translated in a considerable amount of time spent.

Intranet annotations are half-way between local and remote annotations. Intranet annotations are seen as remote annotations for the organization, but local for the rest of the world. These take advantage of the security features of intranets and the easy agreement about models and representation languages supported by local storage systems, while providing the collaborative support typical from remote storage systems. Hybrid annotation storage systems like Annotea may be used both as local and remote annotation systems. Yet, the Annotea system is considerably limited concerning the models and representation languages supported, thus constraining the customization advantages identified for local annotation storage systems.

Whether local or remote storage is chosen, the storage mechanism is an important issue, even if for clients the access process should be transparent and diversified. Relational databases and RDF repositories are commonly used.

I. Browsing, Querying and Filtering

This dimension relates to the support provided to the user manipulation of annotations (*e.g.* searching, browsing, filtering, reading, changing). While some of these tasks are common in many tools (*e.g.* web browsers, text editors) for standard documents, when considering the manipulation of annotations, the same tools are often not fully capable. Consider for example, some of the most popular web browsers (*e.g.* MS Internet Explorer, Mozilla Firefox and Opera), which do not natively support annotations. Yet, despite the fact that several plugins exist for Firefox (*i.e.* Annozilla) allowing document browsing and annotation attachment, any of them allow annotation-driven browsing. Querying and filtering are normally associated one to the other acting as a browsing complement enabling and easing the browsing process. Annotation-driven querying and filtering exploits annotations attached to documents in order to improve the browsing quality and precision, while reducing the user participation in the process.

In order to further improve the browsing process, the user interests and domain knowledge should be taken into account. For that, specific annotation models should be used, including the use of ontologies. Ontologies are then used by querying and filtering engines to associate user requirements with documents contents, reducing the amount of retrieved documents and improving precision and accuracy.

Because different conceptualizations of the world are possible, many ontologies for the same knowledge domain are used by different information communities. As a consequence, automatic services require the capability to map between different ontologies, and in a semantic meaningfully way manage and relate annotations. Even though much improvement has been achieved, ontology alignment [5] and ontology mapping [16] are fundamental issues in the practical aspects of semantic annotation management [21, 24].

J. Accessibility

Accessibility relates to the publishing characteristics and trustiness. It refers to several features that make this dimension very important in the overall Semantic Annotation process, and especially in the use and application of semantic annotations. Publishing relates to publishing decisions and status. It exploits the trustiness defined attributes in order to constrain the access to the annotations.

Three kinds of annotations exist according to publishing:

- Personal Annotations are those made for private and personal usage only. This kind of annotation has only one viewer in mind, which is the author themself. They are only visible to the author, and cannot be obtained or searched by any other user or machine;

- Public Annotations allow the user to express ideas, feelings and comments to the information community. These annotations become available to anyone or anything and therefore are readable and searchable, but not editable by others. This type of annotations are mostly used by the whole WWW community;

- Collaborative Annotations are shared by the information community, therefore readable, searchable and processable by anything or anyone, allowing editing not only by the author but also by those allowed to edit the annotation. They are called collaborative annotations because they permit several people or machines to work on them. This kind of annotation becomes very useful in small workgroups or research groups when collaboration is needed on a same topic.

Trust attributes include but are not limited to those that constrain the access and validation, such as trusting entities, time of creation and limit of validity, allowed accessing location, time periods and users. While very important semantic annotations properties, they are very

difficult to specify and guarantee by the annotation process, supporting mechanisms and further levels of application of the semantic annotations. In fact, trust is still an open issue of the semantic web technology architecture and therefore scarcely supported by currently annotation tools.

K. Evolution, Versioning

Evolution is the process whereby the annotation is maintained coherent according to the document it annotates and (eventually) with the specification mechanisms used (*e.g.* representation language, ontology).

Alternatively, versioning is related to the changes occurring to the content of the annotation. The annotation content may be different (different knowledge about the document) even if the document content, the specification language and the ontology used to represent it are the same. Often, the evolution of annotation implies versioning of annotations, but not the inverse.

L. Information Extraction Process

The Information Extraction (IE) process is concerned with the method used to identify both the parts of the document to annotate and the content of the annotation itself. The process is characterized by two subdimensions: the method type and the level of automation.

With respect to the method sub-dimension, the following types have been generalized:

- Pattern matching is based on regular expressions. Before searching document content, regular expressions have to be specified so that they can be used by matching techniques. Their usage on web pages constitutes an easy way of retrieving data in the case of stable, well known pages [15]. In many cases, like online libraries or web based repositories where changes are seldom made, this approach can be used, as it is quite simple to implement. If the pages on which this technique is to be used change quite regularly, other approaches should be considered. LP^2 [3] is a process that uses pattern matching;

- Natural Language Processing (NLP) methods are able to identify parts-of-speech (*e.g.* verb, noun, pronoun, adjective, adverb, preposition, conjunction, interjection), parts-of-sentence (*e.g.* subject, predicate, object, complements) and linguistic features of the document content. NLP techniques are able to complement the pattern matching methods, improving and disambiguating the meaning of lexicons in the document [7];

- Layout Processing exploits the document layout for information extraction. Quite often, documents of the same business or domain of knowledge have similar layouts, even if their internal structural code is different. For example, online banks all tend to have a web page with a tabled content that show an account balance on the same line as the account number. The layout processing approach "allows to determine if certain pieces of information are visually located above or under, right or left, in column or in row [...] of another piece of information" [15] so that information can be extracted from those layout combinations. These types of method therefore have the ability to extract structural information from the document, allowing the creation of interrelations between distinct parts, which eventually complement the previously extracted information;

- HTML Source Processing is a special case of pattern matching. HTML pages are a special case of document, whose content is formatted by mark ups. However, these mark ups are often semantically meaningless for the document content. Extracting their semantics can provide some information that plain text or layout approaches often do not. This method can take advantage of mark up by contextualizing the pattern matched content;

- Rule-based methods are those that define the set of conditions (LHS[1])to perform a specific action (RHS). The data fulfilling the conditions may be used as parameters for the action, thus acting in accordance to the facts exploited in the conditions. The conditions are based on finer granular elements provided by pattern matching, NLP and layout processing methods, thus forming a more complex but also more configurable and customizable system than the previous;

- Ontology-based methods aim to identify instances of ontology-defined concepts included in the document. It performs the analysis of documents based on previous methods and creates the instances of concepts from the target ontology, into a repository. Gate [1] exploits pattern matching, NLP and rule-based methods into an ontology-based IE system [15];

- Machine learning methods are not a method by itself, but a technique used to deploy systems that are configurable and customizable according to specific IE requirements, document content and layout, business or domain of knowledge. According to a training corpus, the IE method defines the parameters that make the system perform the most similar as possible with the user that produced the training corpus.

[1]Left/Right Hand Side

Machine learning techniques are applied in all the types of methods described above.

The automation degree of the IE process relates to the method type. While pattern matching methods are easily automated, NLP and ontology-based methods are harder to automate. The automation of the method ranges from pure manual (*e.g.* Annozilla) to fully automated (*e.g.* Amilcare, Gate).

Mark up/visualization oriented annotations typically require a low-level of automation. Instead, because automation of the web processing is one of the main goals associated with the Semantic Web, it requires semantically rich data attached to the web pages. Due to the huge amount of information spread along numerous web servers, the web pages annotation should be highly automated. However, fully automated annotations are often incomplete or ambiguous [20] thus requiring user supervision in some domains of knowledge.

So far, the semantic annotation manual process is analogous to the one used by people when reading physical hard-copy documents. First of all, when a term, set of words, sentence or passage needs to be annotated, the reader selects it using the available tools in the application and subsequently annotates it, which mean that it is assigning some new information to the objects previously selected from the text. When desired or suggested by an annotation tool, the annotator can associate the terms to a concept ontology so that it can have some semantic meaning. Manual annotations therefore, not only semantically enrich the text with very accurate annotations as they also express the authors experiences, feelings and thoughts. In addition, these annotations represent much more than meta-data about the text, sometimes even becoming more important than the original text, because they represent knowledge about something of the author point of view.

Fully automatic annotations instead are obtained through different mechanisms that do not require human participation. Unlike manual annotations, automatic approaches still can not incorporate personal reflections into annotations but can also enrich a text semantically, either using Natural Language Processing, Machine Learning or any other technique. Automatic processes normally produce annotations which act like a document meta-data.

Still, semi-automatic approaches take advantages and disadvantages of both manual and automatic annotations. Sharing the same opinions with other research groups [20] "semi-automatic way of annotating content is [...] the most promising method suitable for annotating narratives" means that this mixture of human and machine reliability tend to remove annotations ambiguity and also permit users to include personal and important information to the annotation that the information extraction can not. How far this semi-automatic process can reach in the scope of the semantic web is however an open issue.

IV. Conclusions

Semantic Annotations, as mentioned, are becoming more common, available and usable in conjunction with Semantic Web. Until now, all these interest areas have been relatively new and therefore different terms, that often mean the same, are used by different researchers and research groups.

Some explanations of *annotation process*, *annotation* and *semantic annotation object* are, what their purpose, why they should be used and by whom, have been mentioned in the article in order to give an introduction to those that are not so familiar with those terms.

Several dimensions related to annotation have been identified and described along with their different approaches and terms. Despite not all existing dimensions of annotation have been addressed in this paper, these are some that contribute more for the characterization of the knowledge domain and that require more attention in the near future.

Despite high automation of the annotation process is required in many application domains, it is important to notice that annotations often require the participation and know-how of the user, thus maintaining a certain level of genuine person knowledge that is difficult to capture by fully automatic systems.

This knowledge to be validated, shared and understood by different information communities has to be conceptually categorized. Therefore, so that semantic annotations can have real semantics, the usage of ontologies on annotations is envisaged as a key element in the annotation knowledge domain. However, dealing with ontologies is often a hard task, as problems like ontology alignment and mapping might occur. These are less evident when developing or testing on closed environments, but in open worlds like the (semantic) web, they become more serious and relevant problems. Semantics sharing and ontology mapping is therefore also addressed as a subject of high importance in future research in this field.

Acknowledgements

This work is partially supported by the Portuguese MCT-FCT project EDGAR (POSI/EIA/61307/2004). Thanks to Owen Gilson for his revisions of the paper.

V. References

[1] K Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10(3/4):349–373, 2004.

[2] Sergey Brin, Lawrence Page, Motwani R., and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Report, 1998.

[3] Fabio Ciravegna. $(LP)^2$, an Adaptive Algorithm for Information Extraction from Web-related Texts. Seattle (WA), USA, Aug 2001. Seattle (WA), USA.

[4] Fabio Ciravegna, Alexiei Dingli, Yorick Wilks, and Daniela Petrelli. Amilcare: Adapative Information Extraction for Document Annotation. pages 367–368, Tampere, Finland, 2002. Tampere, Finland, ACM Press.

[5] Marc Ehrig, Steffen Staab, and York Sure. Qom - quick ontology mapping. volume 3298 of *Lecture Notes in Computer Science*, pages 683–697, Hiroshima, Japan, Nov 2005. Hiroshima, Japan, Springer.

[6] Jérôme Euzenat. Eight Questions about Semantic Web Annotations. *IEEE Intelligent Systems*, 17(2):55–62, 2002.

[7] Luca Gilardoni, Chistian Biasuzzi, Massimo Ferraro, Roberto Fonti, and Piercarlo Slavazza. LKMS - A Legal Knowledge Management System Exploiting Semantic Web Technologies. volume 3729 of *Lecture Notes In Computer Science*, pages 872–886, Galway, Ireland, 2005. Galway, Ireland, Springer.

[8] Christian Halaschek-Wiener, Jennifer Golbeck, Andrew Schain, Michael Grove, Bijan Parsia, and Jim Hendler. PhotoStuff - An Image Annotation Tool for the Semantic Web. Galway, Ireland, 2005. Galway, Ireland.

[9] Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-CREAM - Semi-automatic CREAtion of Metadata. volume 2473 of *Lecture Notes in Computer Science*, pages 358–372, Siguenza, Spain, Sep 2002. Siguenza, Spain, Springer.

[10] Siegfried Handschuh, Raphael Volz, and Steffen Staab. Annotation for the Deep Web. *IEEE Intelligent Systems (special issue on information integration)*, 18(5):42–48, 2003.

[11] Laura Hollink, S. Little, and J. Hunter. Evaluating the Application of Semantic Inferencing Rules to Image Annotation. pages 91–98, Banff (Alberta), Canada, 2005. Banff (Alberta), Canada, ACM Press.

[12] José Kahan and Marja-Riitta Koivunen. Annotea: an open RDF infrastructure for shared Web annotations. pages 623–632, Hong Kong, 2001. Hong Kong.

[13] Mozdev. Annozilla. Electronic Citation, 2006.

[14] Questia. Questia. Electronic Citation, 2006.

[15] L. Rodrigo, Richard Benjamins, J. Contreras, D. Patón, D. Navarro, R. Salla, M. Blázquez, P. Tena, and I. Martos. A Semantic Search Engine for the International Relation Sector. volume 3729 of *Lecture Notes in Computer Science*, pages 1002–1015, Galway, Ireland, 2005. Galway, Ireland, Springer.

[16] Nuno Silva and João Rocha. Multi-Dimensional Service-Oriented Ontology Mapping. *International Journal of Web Engineering and Technology*, 2(1):50–80, 2005.

[17] Alan F. Smeaton, Colum Foley, and McDonald Kieran. ANNODEX-ing Broadcast TV News for Semantic Browsing and Retrieval. Galway, Ireland, 2005. Galway, Ireland.

[18] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., 1999.

[19] UDDI. Universal Description, Discovery and Integration. Electronic Citation, 2006.

[20] Jan Uhlir and Martin Falc. Annotating Narratives Using Ontologies and Conceptual Graphs. pages 84–88. IEEE Computer Society, 2004.

[21] Maria Vargas-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. MnM: Ontology Driven Tool for Semantic Markup. Lyon, France, Jul 2002. Lyon, France.

[22] W3C. About local annotations. Electronic Citation, 2006.

[23] W3C. RDF - Resource Description Framework. Electronic Citation, 2006.

[24] The Deep Web. The Deep Web. Electronic Citation, Jul 2005.